

# **LÓGICA BINARIA**

## *OPERACIONES LÓGICAS CON VARIABLES BINARIAS*

**Edición: 100606**

**Luis González**  
**Profesor de Tecnologías de la Información**  
**Departamento de Tecnología**  
**I.E.S. Santa Eugenia**

## LÓGICA BINARIA

Una de las funciones de la Unidad Aritmético Lógica (**ALU**), situada en el núcleo de un procesador es la de realizar las operaciones lógicas con los datos contenidos en una instrucción del programa. Por otra parte, la mayoría de los sistemas automáticos de control deben hacer operaciones lógicas, con las variables proporcionadas por los sensores, para manejar el comportamiento de sus actuadores. Pero, ¿qué es una operación lógica?

### I. OPERACIONES LÓGICAS

Una operación lógica asigna un valor (**CIERTO** o **FALSO**) a cada una de las posibles combinaciones de estado (**CIERTO** o **FALSO**) de las variables que intervienen en la operación. Las variables que intervienen en una operación lógica sólo pueden tomar uno de estos dos valores, **cierto** o **falso**. Y el resultado de una operación lógica puede ser, tan sólo, **cierto** o **falso**.

Por ejemplo, imagínate el sistema de control del toldo de una cafetería, que se gobierna mediante una operación lógica. Para que el motor que extiende el toldo se accione deberá tener en cuenta dos factores: ¿es de día? ¿está lloviendo? Si estos dos factores son ciertos, el motor debe ponerse en marcha y extender el toldo.

De día	Llueve	Activar motor
Falso	Falso	Falso
Falso	Cierto	Falso
Cierto	Falso	Falso
Cierto	Cierto	Cierto

Los resultados de una operación lógica, para cada uno de los valores posibles de las variables, se fijan en una tabla denominada **Tabla de Verdad**, como la del ejemplo anterior.

A diferencia de las operaciones lógicas, una operación aritmética asigna un valor numérico (*por ejemplo: 2,52*) como resultado de aplicar un algoritmo de cálculo (*por ejemplo: producto  $a \times b$* ) a los valores de las variables (*por ejemplo:  $a=7,2$  y  $b=0,35$* ). Cada una de las variables que intervienen en una operación aritmética puede tomar un valor numérico cualquiera y las operaciones aritméticas más importantes son: suma, resta, producto, cociente y raíz cuadrada.

Las operaciones lógicas más importantes son: **EQUAL** (idéntico), **NOT** (negación), **OR** (O), **AND** (Y), **NOR** (O negada), **NAND** (Y negada), **OREX** (O exclusiva) y **NOREX** (O exclusiva negada). Veamos con detalle estas operaciones:

### II. FUNCIÓN IGUAL (EQUAL)

El resultado **S** de aplicar la función lógica **equal**, sobre una variable **a**, es muy simple: **si  $a$  es CIERTO (1)  $S$  es CIERTO (1) y, si  $a$  es FALSO (0),  $S$  es FALSO (0)**. Estas dos resultados posibles se muestran en la tabla de verdad adjunta:

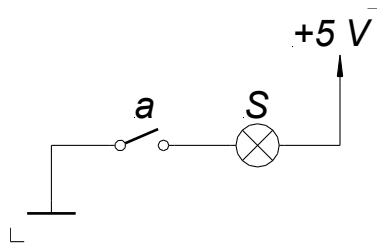
<b>a</b>	<b>S</b>
1	1
0	0

Tabla de verdad de la función IGUAL

Y la función lógica la escribimos así:  $S = a$

Un ejemplo sencillo de aplicación práctica de esta función lógica sería el encendido de las luces del alumbrado público. En algún lugar de la ciudad se instala un detector crepuscular, que detecta cuándo es de noche y controla un interruptor que enciende las luces de las calles: si es de noche (1) se encienden las lámparas (1); si NO es de noche (0) NO se encienden las lámparas (0).

Un circuito eléctrico capaz de implementar la función lógica IGUAL es el siguiente:



Circuito eléctrico que implementa la función IGUAL

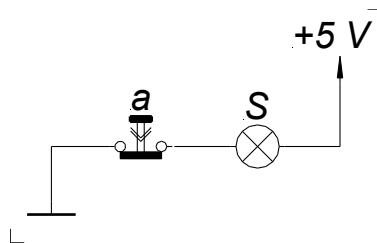
### III. FUNCIÓN NO (NOT)

El resultado **S** de aplicar la función lógica **NOT**, sobre una variable **a**, es muy simple: si **a** es **CIERTO (1)** **S** es **FALSO (0)** y, si **a** es **FALSO (0)**, **S** es **CIERTO (1)**. Estas dos resultados posibles se muestran en la tabla de verdad adjunta. Se conoce también como función negación: **S** equivale a **a** negada.

<b>a</b>	<b>S</b>
1	0
0	1

Figura 3: Tabla de verdad de la función NO

Un ejemplo sencillo de aplicación práctica de esta función lógica sería el circuito que controla el acceso a una oficina bancaria, a través de una puerta automática equipada con un detector de metales que cierra un interruptor. Si el detector de metales **SI** nota que el cliente lleva objetos metálicos (1) y la puerta **NO** se abre (0); en cambio, si el cliente **NO** lleva objetos metálicos (0), la puerta **SI** se abre (1).



Circuito eléctrico que implementa la función NO

La función NO, conocida también como **inversión** o **negación**, se escribe así:  $S = \bar{a}$ . Al aplicarla sobre una variable binaria obtenemos la palabra invertida. Por ejemplo:

Si  $a = 0xf85c$ , es decir: **1111 1000 0101 1100**

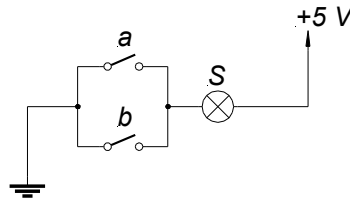
entonces  $S = \bar{a} = 0x07a3$ , es decir: **0000 0111 1010 0011** Compruébalo

### IV. FUNCIÓN O (OR)

La función **OR** equivale a la **conjunción disyuntiva O**. El resultado **S** de aplicar la función lógica **OR**, sobre dos variables **a** y **b** es el siguiente: **S** es cierto si **a** es **CIERTO (1)** o si **b** es **CIERTO (1)**.

Una aplicación práctica sencilla de la operación lógica **OR**, sería el circuito de señalización instalado en un comercio, en el que se puede entrar por dos puertas distintas, que avisaría al dependiente al entrar un cliente por cualquiera de las dos puertas del establecimiento. Si un cliente entra por la puerta **a** (1) **O** si un cliente entra por la puerta **b** (1), el timbre suena (1). Si no entra ningún cliente por ninguna de las puertas a (0) ni b (0). El timbre NO suena (0).

Un circuito eléctrico compuesto por dos interruptores en paralelo, cumple la lógica **OR**. La lámpara **SI** se encenderá (1) si se acciona el interruptor **a** (1) **O** si se acciona el interruptor **b** (1) **O** si se accionan ambos interruptores. Si no se acciona ningún interruptor, la lámpara **NO** se encenderá (0).



*Circuito eléctrico que implementa la función OR*

Cuando se aplica una operación lógica sobre 2 variables caben 4 combinaciones posibles. Los resultados de la operación lógica OR, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

<b>a</b>	<b>b</b>	<b>S</b>
0	0	0
0	1	1
1	0	1
1	1	1

*Tabla de verdad de la función lógica O (OR)*

La función lógica que se corresponde con esta tabla de verdad, y que recibe el nombre de **suma lógica**, se escribe del siguiente modo:

$$S = a + b$$

Veamos cómo se aplica esta función lógica a sendas variables binarias, por ejemplo:

Si  $a = 0x7ab5$ , es decir: 0111 1010 1011 0101

y  $b = 0xe013$ , es decir: 1110 0000 0001 0011

entonces  $S = a + b = 0xfab7$ , es decir: 1111 1010 1011 0111 Compruébalo

## V. FUNCIÓN Y (AND)

La función **AND** equivale a la **conjunción copulativa Y**: El resultado **S** de aplicar la función lógica **AND**, sobre dos variables **a** y **b** es el siguiente: **S es CIERTO si a es CIERTO (1) Y si b es CIERTO (1)**.

Los resultados de la operación lógica AND, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta.

<b>a</b>	<b>b</b>	<b>S</b>
0	0	0
0	1	0
1	0	0
1	1	1

*Tabla de verdad de la función lógica Y (AND)*

Una aplicación de la operación lógica AND, sería el sistema de control de los pasajeros en un aeropuerto. Cada pasajero debe pasar por tres controles: ¿Tiene tarjeta de embarque? ¿Tiene pasaporte en regla? ¿No lleva objetos metálicos peligrosos? Una empleada del aeropuerto comprueba que tiene un billete válido y le da una tarjeta de embarque; a continuación, un agente de policía verifica que su pasaporte está en regla y no está en la lista de personas reclamadas y, finalmente, un grupo de agentes comprueban su equipaje de mano con un escáner y un arco detector de metales. Un pasajero sólo puede embarcar en el avión si tiene tarjeta de embarque (1)

Y su pasaporte está en regla (1) Y no lleva consigo objetos peligrosos (1). En los demás casos no puede embarcar.

Es fácil construir un circuito eléctrico que cumple la lógica **AND**: dos interruptores en serie, **a** y **b**, por ejemplo. La lámpara **S** se encenderá tan sólo si se actúa sobre el interruptor **a** (1) **Y** sobre el interruptor **b** (1). En todos los demás casos, la lámpara **NO** se encenderá.

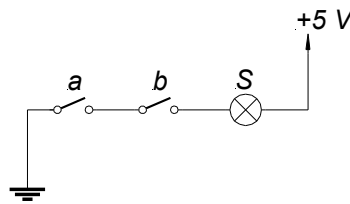


Figura 8: Circuito eléctrico que implementa la función AND

La función lógica que se corresponde con esta tabla de verdad, y que recibe el nombre de **producto lógico**, se escribe del siguiente modo:

$$S = a \cdot b$$

Veamos cómo se aplica esta función lógica a sendas variables binarias, por ejemplo:

Si  $a = 0x2d84$ , es decir: 0010 1101 1000 0100

y  $b = 0x7f9b$ , es decir: 0111 1111 1001 1011

entonces  $S = a \cdot b = 0x2d80$ , es decir: 0010 1101 1000 0000 Compruébalo

## VI. FUNCIÓN O NEGADA (NOR)

La función **NOR** equivale a la función **OR negada** o, dicho de otro modo, equivale a la **conjunción copulativa NI**. El resultado **S** de aplicar la función lógica **NOR**, sobre dos variables **a** y **b** es el siguiente: **S es CIERTO si a es FALSO (0) y si b es FALSO (0)**.

Una aplicación práctica sencilla de la operación lógica **NOR**, sería el sistema de seguridad de un puente levadizo. Un detector **a** se activa cuando entra un vehículo en el puente, por el carril derecho. Otro detector **b** se activa cuando entra otro vehículo por el carril contrario. Los motores que accionan el sistema de elevación del puente sólo deben ponerse en marcha si se da la condición **NOR**: no hay ningún vehículo circulando por el carril derecho NI por el carril izquierdo.

Un circuito eléctrico compuesto por dos interruptores normalmente cerrados, en serie, cumple la lógica **NOR**: la lámpara **SI** se encenderá (1) si **NO** se acciona el interruptor **a** (0) **NI** se acciona el interruptor **b** (0). Si se acciona cualquiera de los dos interruptores, la lámpara **NO** se encenderá (0).

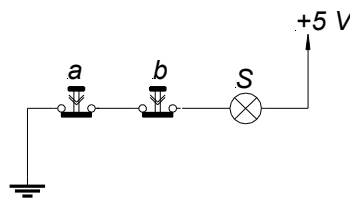


Figura 10: Circuito eléctrico que implementa la función NOR

Los resultados de la operación lógica **NOR**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

<i>a</i>	<i>b</i>	<i>S</i>
0	0	1
0	1	0
1	0	0
1	1	0

Tabla de verdad de la función lógica O NEGADA (NOR)

La función lógica que se corresponde con esta tabla de verdad se escribe del siguiente modo:

$$S = \overline{(a + b)} \quad \text{y, si te fijas en la tabla de verdad, equivale a} \quad S = \bar{a} \cdot \bar{b}$$

Veamos cómo se aplica esta función lógica a sendas variables binarias, por ejemplo:

Si  $a = 0x8a34$ , es decir: 1000 1010 0011 0100

y  $b = 0x25a1$ , es decir: 0010 0101 1010 0001

entonces  $S = \overline{(a + b)} = 0x504a$ , es decir: 0101 0000 0100 1010 Compruébalo

## VII.FUNCIÓN Y NEGADA (NAND)

La función **NAND** equivale a la función **Y negada**. El resultado **S** de aplicar la función lógica **NAND**, sobre dos variables **a** y **b** es el siguiente: **S es CIERTO si a es FALSO (0) o si b es FALSO (0) o si son FALSAS ambas variables**. Los resultados de la operación lógica **NAND**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

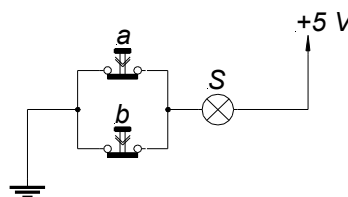
<i>a</i>	<i>b</i>	<i>S</i>
0	0	1
0	1	1
1	0	1
1	1	0

Tabla de verdad de la función lógica Y NEGADA (NAND)

Una aplicación práctica sencilla de la operación lógica **NAND** sería, por ejemplo, el control del aire acondicionado de un edificio inteligente. Supongamos que el edificio está equipado con un detector crepuscular, que se activa al llegar la noche. Durante el día el detector está desactivado (0) y durante la noche el detector está activado (1).

Supongamos también que en la entrada del edificio hay un sistema de recuento de personas que se pone a **CERO** (0) cuando hay alguien en el edificio y se pone a **UNO** (1) cuando todo el mundo ha salido ya. ¿Cómo controlar la puesta en marcha del aire acondicionado? Muy fácil, con un circuito que siga la lógica **NAND**: el aire acondicionado se parará cuando sea de noche y no quede nadie en el edificio.

Un circuito eléctrico compuesto por dos interruptores normalmente cerrados, en paralelo, cumple la lógica **NAND**: la lámpara **SI** se encenderá (1) si **NO** se acciona el interruptor **a** (0) o si **NO** se acciona el interruptor **b** (0) o si **NO** se accionan ambos interruptores.



Circuito eléctrico que implementa la función NAND

La función lógica que se corresponde con esta tabla de verdad se escribe del siguiente modo:

$$S = \overline{(a \cdot b)} \text{ y, si te fijas en la tabla de verdad, equivale a } S = \bar{a} + \bar{b}$$

Veamos cómo se aplica esta función lógica a sendas variables binarias, por ejemplo:

Si  $a = 0x3fb7$ , es decir: 0111 1111 1011 0111  
 y  $b = 0xd75b$ , es decir: 1101 0111 0101 1011  
 entonces  $S = \overline{(a \cdot b)} = 0xa8ec$ , es decir: 1010 1000 1110 1100 Compruébalo.

## VIII.FUNCIÓN O EXCLUSIVA (OREX)

La función **OREX** se conoce también con los nombres de **O EXCLUSIVA** y **XOR**. El resultado **S** de aplicar la función lógica **OREX**, sobre dos variables **a** y **b** es el siguiente: **S es CIERTO solo si a es CIERTO (1) o si b es CIERTO (1), pero no si ambas variables son ciertas**. Los resultados de la operación lógica **OREX**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

Tabla de verdad de la función lógica O EXCLUSIVA (OREX)

Si te fijas, la función OREX es cierta cuando ambas variables son distintas

Un circuito eléctrico como el del esquema siguiente, compuesto por dos pulsadores dobles **NA + NC**, cumple la lógica **OREX**: la lámpara **S** se encenderá (1) **EXCLUSIVAMENTE** si se acciona el pulsador **a** o si se acciona el pulsador **b**, pero **NO** se encenderá si se accionan simultáneamente ambos pulsadores. Tampoco se encenderá si no se acciona ninguno de los dos pulsadores.

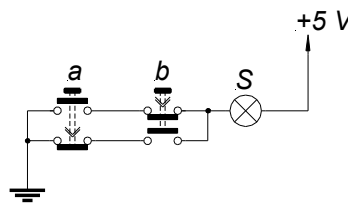


Figura 14: Circuito eléctrico que implementa la función OREX

La función lógica que se corresponde con la tabla de verdad de la función OREX es esta:

$$S = \bar{a} \cdot b + a \cdot \bar{b}$$

Veamos cómo se aplica esta función lógica a sendas variables binarias, por ejemplo:

Si  $a = 0xb47d$ , es decir: 1011 0100 0111 1101  
 y  $b = 0xd75b$ , es decir: 1101 0111 0101 1011  
 entonces  $S = \bar{a} \cdot b + a \cdot \bar{b} = 0x6326$ , es decir: 0110 0011 0010 0110 Compruébalo.

## IX. FUNCIÓN NOREX

La función **NOREX** se conoce también con el nombre de **OR EXCLUSIVA NEGADA**. El resultado **S** de aplicar la función lógica **NOREX**, sobre dos variables **a** y **b** es el siguiente: **S es CIERTO si a y b son ciertos O si a y b son falsos. Es decir, si ambas variables tienen el mismo valor.** Los resultados de la operación lógica **NOREX**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

<b>a</b>	<b>b</b>	<b>S</b>
0	0	1
0	1	0
1	0	0
1	1	1

Tabla de verdad de la función lógica **O EXCLUSIVA NEGADA (NOREX)**

Si te fijas, la función **NOREX** es cierta cuando ambas variables son **iguales**

Un circuito eléctrico como el del esquema siguiente, compuesto por dos pulsadores dobles **NA + NC**, cumple la lógica **NOREX**: la lámpara **S** se encenderá si se accionan **ambos pulsadores o si no se acciona ninguno de ellos**.

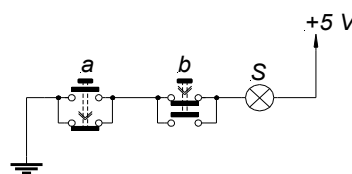


Figura 16: Circuito eléctrico que implementa la función **NOREX**

La función lógica que se corresponde con la tabla de verdad de la función **NOREX** es esta:

$$S = \bar{a} \cdot \bar{b} + a \cdot b$$

Veamos cómo se aplica esta función lógica a sendas variables binarias, por ejemplo:

Si  $a = 0xa39b$ , es decir:

1010 0011 1001 1011

y  $b = 0x5f0c$ , es decir:

0101 1111 0000 1100

entonces  $S = \bar{a} \cdot \bar{b} + a \cdot b = 0x0368$ , es decir:

0000 0011 0110 1000

Compruébalo.

Otro ejemplo de aplicación de la función lógica **NOREX** es la corrección automática de textos: si una persona escribe una palabra en su procesador de textos, el corrector ortográfico la comparará con todas las palabras semejantes de su diccionario aplicando una función **NOREX** entre ellas. Si alguna letra no coincide, detectará que hay un error, porque el resultado de la función **NOREX** no entrega unos en todos los bit. Por ejemplo, si escribimos con una falta de ortografía la palabra **lobo**:

<b>Palabra</b>	<b>Código ASCII</b>	<b>Código binario</b>
lovo	6C6F766F	01101100011011110111011001101111
lobo	6C6F626F	01101100011011110110001001101111
Función NOREX:		11111111111111111111010111111111

de este modo, el procesador es capaz de detectar que el error está en el tercer carácter.

Luis González  
Profesor de Tecnología